R source with main MCMC function

```
#################### mcmc.R ################################################

mcmc <- function(gDNA, cDNA, tissue, ugrid, nsave, nskip, hyper )
 {
  # inital state
        n <- length(gDNA)  ##number of tumors
        K <- nrow(ugrid)
        pi <- rep( 1/K, K )            ## simplex distribution for U's
        theta <- c(50,50,5,5,25,25)  ## 4 means and 2 variances (one cDNA and one gDNA varianc
        mu.mat <- cbind( theta[1:2], theta[3:4], theta[c(1,4)] )
        sig2.mat <- cbind( theta[5:6], theta[5:6], theta[(5:6)] )
        id.norm <- nrow(ugrid)  ### (1,0,0) = normal tissue case



  # run parameters
        nscan <- nskip*nsave ## number of scans
        skipcount <- 0
        isave <- 1

  # storage info
        pisave <- matrix(NA, nsave,K)     ## storage for pi's
        thetasave <- matrix(NA, nsave,6)  ## storage for theta's

  # hyper-parameters
        alpha <- hyper$alpha    ## total mass of Dirichlet prior on pi
        sig0 <- hyper$sig0      ## prior guess at measurement standard deviation
        n0 <- hyper$n0       ## like prior sample size on variance

  ## run
  for( iscan in 1:nscan )
   {
    skipcount <- skipcount+1
    #####################################################################
    #
    # update U's [these are the mixing proportions over the three states
    # Do this by Gibbs

    mu <- mu.mat %*% t(ugrid)   ## 2 x ngrid  , rows for gDNA, cDNA means given U
    sdev <- sqrt( sig2.mat %*% t(ugrid)^2 ) ## 2 x ngrid,rows for gDNA, cDNA variance given U
                        ## note gDNA independent of cDNA given U, and both normal
    xx2 <- -0.5*( outer( mu[1,], gDNA, "-" )/sdev[1,] )^2
    yy2 <- -0.5*( outer( mu[2,], cDNA, "-" )/sdev[2,] )^2

    # ngrid x n
    logp.gDNA  <- xx2 - log( sdev[1,] )
logp.cDNA  <- yy2 - log( sdev[2,] )

logp <- logp.gDNA + logp.cDNA +  log(pi)
mm <- apply(logp,2,max)
foo <- exp( t(logp) - mm  )
foo.s <- apply(foo,1,sum)
pp <- foo/foo.s
cpp <- t( apply( pp, 1, cumsum ) )  ## cumulative dist of each U over grid
uu <- runif( n )
bar <- cpp > uu
U.id<- apply( bar, 1, which.max )  ## the first ‘1’
U.id[tissue=="Normal"] <- id.norm   ## use the known label for normal tissue
U  <- t( ugrid[U.id,] )         ## tumor sample mixing rates


######################################################################
# Update theta:
# first the two horizontal means [...theta[1] and theta[3]...]

# compute the statistics for the full conditionals
sig2 <- sig2.mat %*% U^2  ## 2 x n  , rows for gDNA, cDNA variance given U
A1 <- sum( (1/sig2[1,])*(U[1,]+U[3,])*gDNA )
A2 <- sum( (1/sig2[1,])*(U[2,])*gDNA )
B1 <- sum( (1/sig2[1,])*(U[1,]+U[3,])^2 )
B2 <- sum( (1/sig2[1,])*(U[2,])^2 )
C <- sum( (1/sig2[1,])*(U[1,]+U[3,])*(U[2,]) )
mean.1 <- (A1*B2-A2*C)/(B1*B2-C^2)
mean.2 <- (A2*B1-A1*C)/(B1*B2-C^2)
v.1 <- B2/(B1*B2-C^2)
v.2 <- B1/(B1*B2-C^2)
rho <- -C/(B1*B2-C^2)
#Gibbs (from the bivariate normal posterior)
theta[1] <- rnorm(1, mean=mean.1, sd=sqrt(v.1) )
csd <- sqrt( B1*(1-C^2/(B1*B2))/(B1*B2-C^2) )
theta[3] <- cnorm(mu=(mean.2-C*(theta[1]-mean.1)/B2), sigma=csd )
# same thing (could be in parallel) for cDNA means theta[2], theta[4]
A1 <- sum( (1/sig2[2,])*(U[1,])*cDNA )
A2 <- sum( (1/sig2[2,])*(U[2,]+U[3,])*cDNA )
B1 <- sum( (1/sig2[2,])*(U[1,])^2 )
B2 <- sum( (1/sig2[2,])*(U[2,]+U[3,])^2 )
C <- sum( (1/sig2[2,])*(U[2,]+U[3,])*(U[1,]) )
mean.1 <- (A1*B2-A2*C)/(B1*B2-C^2)
mean.2 <- (A2*B1-A1*C)/(B1*B2-C^2)
v.1 <- B2/(B1*B2-C^2)
v.2 <- B1/(B1*B2-C^2)
rho <- -C/(B1*B2-C^2)
#Gibbs (from the bivariate normal posterior)
theta[2] <- rnorm(1, mean=mean.1, sd=sqrt(v.1) )
```

```
csd <- sqrt( B1*(1-C^2/(B1*B2))/(B1*B2-C^2) )
theta[4] <- cnorm(mu=(mean.2-C*(theta[2]-mean.1)/B2), sigma=csd )

# update matrix version
mu.mat <- cbind( theta[1:2], theta[3:4], theta[c(1,4)] )


## now the variances (one for gDNA theta[5], and one for cDNA theta[6], common to componen

## A Gibbs sample, using an inverse chi-square prior
shape.g <- (n0+n)/2
tmp.mean <- (U[1,] + U[3,])*theta[1] + U[2,]*theta[3]
bar <- U[1,]^2 + U[2,]^2 + U[3,]^2
tmp.stat <- sum( (gDNA -tmp.mean)^2/bar  )
rate.g <- (1/2)*( n0*sig0^2 + tmp.stat  )
theta[5] <- 1/rgamma(1,shape=shape.g,rate=rate.g)

shape.c <- (n0+n)/2
tmp.mean <- U[1,]*theta[2] + (U[2,]+U[3,])*theta[4]
tmp.stat <- sum( (cDNA -tmp.mean)^2/bar )
rate.c <- (1/2)*( n0*sig0^2 + tmp.stat )
theta[6] <- 1/rgamma(1,shape=shape.c,rate=rate.c)

# update matrix version
sig2.mat <- cbind( theta[5:6], theta[5:6], theta[5:6] )


######################################################################
# Update pi  by Gibbs (ignore the normals...)

    tmp <- table(U.id[!(tissue=="Normal")])
    cnts <- rep(0,K)    ## the empirical distribution of U's on their grid
    names(cnts) <- 1:K
    cnts[ match(names(tmp),1:K ) ] <- tmp
    gg <- rgamma(K, shape=(cnts+alpha/K) )
    pi <- gg/sum(gg)

######################################################################
#
# Store summary statistics periodically..
if( skipcount == nskip )
 {
  skipcount <- 0
  pisave[isave,] <- pi
  thetasave[isave,] <- theta
    ## maybe some posterior info for each U?
  print( isave )
  isave <- isave+1
```

```
     }

    }
   out <- list( data=cbind(gDNA,cDNA,tissue), mcmc=c(nsave,nskip), hyper=hyper,
                theta=thetasave, pi=pisave, ugrid=ugrid )
   out
  }


# a function to simulate a normal given it is positive
cnorm <- function(mu,sigma,nsim=1)
 {
  u <- runif(nsim)
  tmp <- pnorm( -mu/sigma )
  bar <- u*(1-tmp) + tmp
  foo <- qnorm(bar)
  # a bailout if bar=1 (i.e. if numerical error makes it hard to get the conditioned normal)
  x <- ifelse( foo < Inf,  mu + sigma*foo, sigma*log(1/runif(nsim)) )
  x
 }

#############################################################################
```

R source running posterior computations for Pirc.

```
################## pirc-2.R ##############################################
# Data:
        dat <- read.delim("data1.txt",header=TRUE)
        tissue <- dat[-37,3]  ## 37 is an extreme outlier
        gDNA <- dat[-37,7]    ## replicate-averaged numbers
        cDNA <- dat[-37,10]    ## replicate-averaged numbers

## grid of values supporting U
        load("grid.RData")  ## thresholded 40x40 --> 861 grid points
         ## p1keep, p2keep
        ugrid <- cbind( p1keep, p2keep, 1-p1keep-p2keep )
        ugrid[ugrid<0] <- 0
        ugrid[ugrid>1] <- 1  ## trim some round-off error

source("mcmc.R")  ## the main function

fit <- mcmc( gDNA=gDNA, cDNA=cDNA, tissue=tissue, ugrid=ugrid, nsave=5000,
        nskip=100, hyper=list( alpha=1, sig0=5, n0=1 ) )

save(fit, file="results/fit-pirc-2-long.RData" )
###########################################################################
```

R source to plot the estimated admixture distribution (Pirc shown)

```
############################ plot-pi.R ##########################################

load("results/fit-pirc-2-long.RData") ## 500,000 run
pihat <- apply(fit$pi,2,mean)

load("grid.RData")

ugrid <- fit$ugrid
bary <-t(  T %*% t( ugrid[,2:3] ) + c(1/2, sqrt(3)/2)  )  ## barycentric coordinates

broman <- rev( rainbow(256, start=0, end=2/3 ) )

n <- length(pihat)
cls <-  rev( rainbow(n, start=0, end=2/3) )

ord <- order( pihat )

pdf( file="plots/pihat-pirc.pdf" )
par( mar=c(0,0,2,0) )
plot( bary[ord,1], bary[ord,2], pch=18, col=cls, axes=FALSE, xlab="", ylab="",
      xlim=c(-1/6, 7/6), ylim=c(-1/6, 1+ 1/6), cex=1.6 , main="Pirc: estimated admixture")
eps <- 0
text( 0, -1/16, labels="gLOH/cLOH", cex=1.5 )
text( 1, -1/16, labels="gMOH/cLOH", cex=1.5 )
text( 1/2, sqrt(3)/2+1/16, labels="gMOH/cMOH", cex=1.5 )
dev.off()

## get some summaries
# majority prob

ok1 <- ugrid[,1] > 1/2
ok2 <- ugrid[,2] > 1/2
ok3 <- ugrid[,3] > 1/2
p.maj <- c( sum( pihat[ok1] ), sum(pihat[ok2]), sum( pihat[ok3] ) )
names(p.maj) <- c("gMOH/cMOH", "gLOH,cLOH","gMOH/cLOH" )
foo <- apply(ugrid,1,which.max)
p.plur <- c( sum( pihat[foo==1] ), sum( pihat[foo==2] ), sum( pihat[foo==3] ) )
names(p.plur) <- names(p.maj)
probs.pirc <- cbind( p.maj, p.plur )
dimnames(probs.pirc)[[2]] <- c("Pr( majority )","Pr( plurality )" )

##################################################################################
```